

Curso Inteligência Artificial: do Zero ao Infinito

Modelos em Produção

Universidade Federal de Mato Grosso

Agenda

1 TF Serving

2 gRPC

3 Aplicação

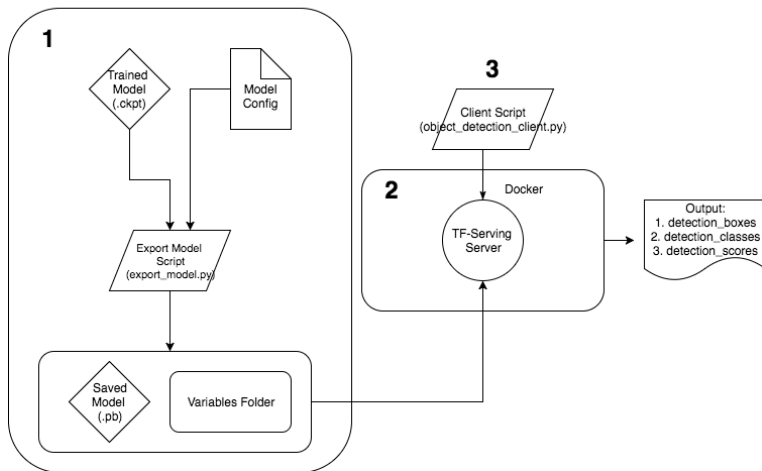
TF Serving

- Podemos disponibilizar um modelo para Object Detection utilizando o framework **TF Serving**.
- TF Serving é um sistema de serviço flexível e de alto desempenho para modelos de aprendizado de máquina, projetado para ambientes de produção.
- Facilita a implantação de novos algoritmos e fornece integração imediata com os modelos do TensorFlow.

Fonte: Documentação Oficial Tensorflow

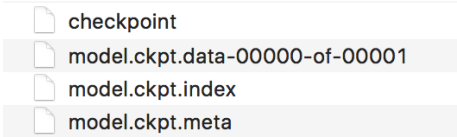
TF Serving

Architecture for serving object detection models using TensorFlow Serving



TF Serving

- Assumindo que você treinou um modelo para *object detection* utilizando o Tensorflow, os *checkpoints* terão a seguinte formato:



```
graph TD; checkpoint --> data["model.ckpt.data-00000-of-00001"]; checkpoint --> index["model.ckpt.index"]; checkpoint --> meta["model.ckpt.meta"];
```

- Você pode utilizar esses arquivos para inferência, no entanto, não são adequados para o ambiente de produção.

Paper: [How to deploy an Object Detection Model with TensorFlow serving](#)

TF Serving

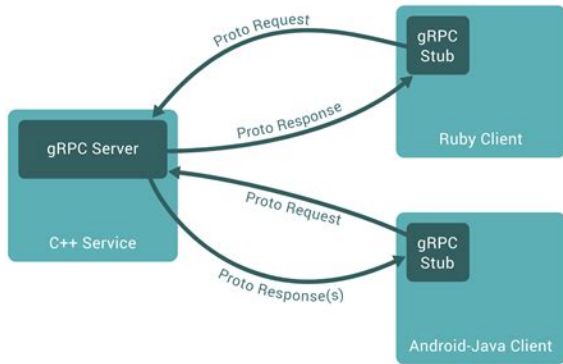
- Podemos converter o modelo para um *frozen graph*, que contém a arquitetura do modelo e os pesos em um único arquivo.
- Para isso, utilize o script *export.py* presente no código de treinamento:

```
$ python export.py \  
  --config_file=configs/parameters.yaml \  
  --pipeline_config_file=pipeline_file.config \  
  --checkpoint_dir=checkpoints_folder \  
  --output_export_dir=exported_checkpoints
```

Código: https://github.com/freds0/fault_detection_power_transmission_lines

- Após converter os checkpoints, podemos "servir" o modelo utilizando o TF Serving.
- O TF Serving utiliza o protocolo gRPC, desenvolvido pelo Google.
- O gRPC permite que um cliente execute uma função em um servidor remoto

Fonte: gRPC - A high performance, open source universal RPC framework



- O gRPC utiliza *Protocol Buffers* para serializar os dados.
- Dessa forma, os dados ficam menores quando comparados com JSON e XML.
- Assim, é necessário definir uma interface que utilize esse protocolo.

Fonte: gRPC - A high performance, open source universal RPC framework

Aplicação

- A aplicação é dividida em *backend* e *frontend*.
- No *backend* tem-se uma aplicação TF-Serving aguardando chamadas na porta 5000.
- No *frontend* tem-se uma aplicação Web rodando na porta 8000.

Código: https://github.com/freds0/flask_fault_detection_power_transmission_lines

Referencias

- Documentação Oficial Tensorflow
 - ▶ <https://www.tensorflow.org/tfx/guide/serving2>
- How to deploy an Object Detection Model with TensorFlow serving
 - ▶ <https://www.freecodecamp.org/news/how-to-deploy-an-object-detection-model-with-tensorflow-serving-d6436e65d1d9>

Curso Inteligência Artificial: do Zero ao Infinito

Modelos em Produção

Universidade Federal de Mato Grosso